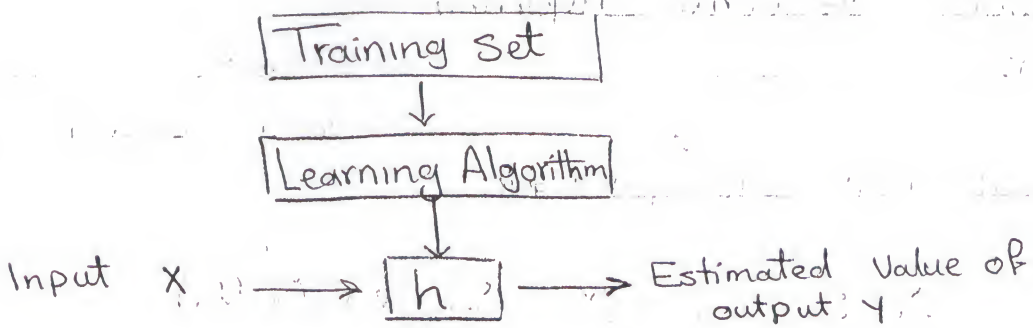
Lec : 4



* Learning Algorithms is used to get Hypothesis

$$h_\theta(x) = \theta_0 + \theta_1 x$$

* The gradient decent algorithm is Learning Algorithm. used to minimize the objective function $\min_{\theta_0, \theta_1} J(\theta_0, \theta_1)$

How

1. Start with some initials θ_0, θ_1
2. Keep changing θ_0, θ_1 to reduce $J(\theta_0, \theta_1)$ until we hope fully end up at a minimum.

example in slide 6:

we start from a θ_0, θ_1 and from these θ_0, θ_1 which we choose to start with, we keep changing them to reduce $J(\theta_0, \theta_1)$ until we reach a minimum point

example in slide 7:

we start from other values of θ_0, θ_1 and follow the same steps until we reach another minimum point for the same figure.

→ each of these minimum points is called local minimum we want to get the best of them (the minimum of them) and called it global minimum which give the minimum value of $J(\theta_0, \theta_1)$

at First we will study the algorith of gradient descent generally
(GDA For any Function)

Gradient descent algorithm

Note: it is a general way to get min value of any Function so we use it to get h_θ $J(\theta$

repeat until Convergence { ← means until end up at a Local minimum.

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

}

↓ Learning rate that controls how big step we take down with gradient decent.

big
we reach to local minimum very fast

small
We reach to local minimum very slow.

So we need to choose suitable α

- we know that $J(\theta_0, \theta_1)$ is Constant For each line so when we Chang θ_0, θ_1 together or change θ_0 only or Chang θ_1 only we get another different Line with new $J(\theta_0, \theta_1)$
- During gradient descent algorithm we must change θ_0, θ_1 together in each Step (at the same time) to get one value to $J(\theta_0, \theta_1$ of only one line .

Correct:

one move

$$\begin{cases} temp0 := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) \\ temp1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1) \\ \theta_0 := temp0 \\ \theta_1 := temp1 \end{cases}$$

] $J(\theta_0, \theta_1)$ For one line θ_0, θ_1 Change in the same time.

Incorrect

2 moves

$$\begin{cases} temp0 := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) \\ \theta_0 := temp0 \\ temp1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1) \\ \theta_1 := temp1 \end{cases}$$

] $J(\theta_0, \theta_1)$ For one line in which θ_0 changes but θ_1 Const

] $J(\theta_0, \theta_1)$ For another line in which θ_0 Constant and θ_1 changes.

- Simplified ($\theta_0 \doteq 0$)

Hypothesis: $h_\theta(x) = \theta_1 x$

Parameters: θ_1

Cost Function: $J(\theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$

Goal: $\min_{\theta_1} J(\theta_1)$ (minimize error to get the Best line)

Gradient descent algorithm: (one of many ways that can be used to get $h_\theta(x)$

→ we use it to minimize error $J(\theta_0, \theta_1)$ to get the best Line $h_\theta(x)$

repeat until Convergence {

$$\theta_1 := \theta_1 - \alpha \frac{d}{d\theta_1} J(\theta_1)$$

}

So → Two Parameters Control GDA (α , $\frac{d}{d\theta_1} J(\theta_1)$) to get minimam $J(\theta_1)$

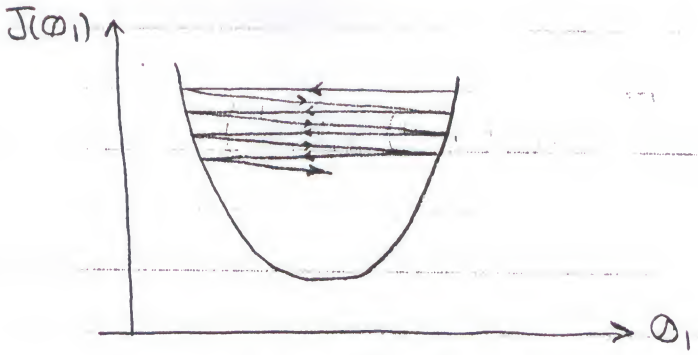
Alpha α

- IF α is too small, gradient descent can be slow.
- IF α is too Large, gradient descent can overshoot the minimum. It may fail to Converge, or even diverge.

- Gradient descent can Converge (تقارب) to local minimum, even with the learning rate α Fixed.
- As we approach a Local minimum, gradient descent will automatically take Smaller steps. So, no need to decrease α over time.

Neglecting the effect of the slope $\left(\frac{d J(\theta_1)}{d \theta_1}\right)$

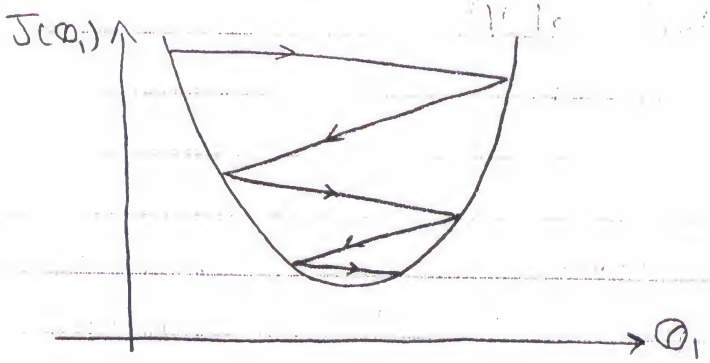
① α is too small



→ the values of θ_1 will move between big and small values (+ve & -ve values of slope)

→ It will reach to Local minimum poit but very slow.

② α is too great



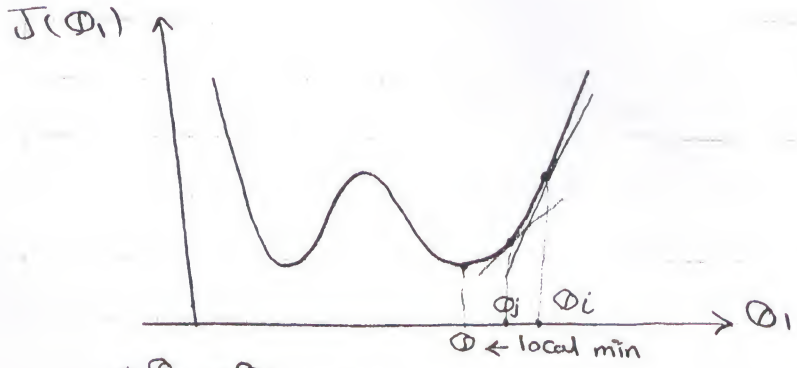
→ The values of θ_1 will move between big and small values (+ve & - ve values of slope)

→ it may not be able to reach to Local minimum because of its Large step.

2 $\frac{d J(\Theta_1)}{d \Theta_1}$ ← Slope (المشتقة الأولى لدالة الخطأ)

* Assuming That α is Fixed

① Slope = + Ve Value



$$\Theta_{1_{new}} := \Theta_{1_{old}} - \alpha \frac{d J(\Theta_1)}{d \Theta_1}$$

→ α Fixed

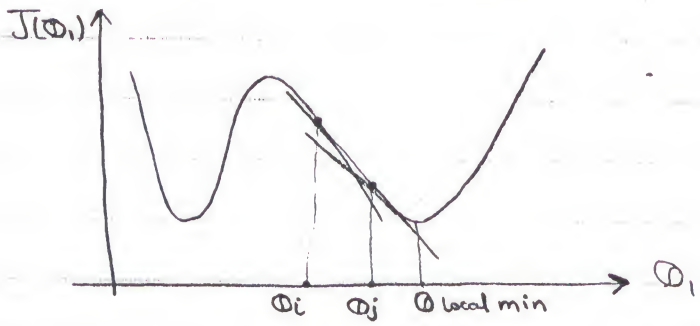
→ $\Theta_{1_{new}}$ dependes on $\frac{d J(\Theta_1)}{d \Theta_1}$

* $\Theta_{1_{old}} = \Theta_i$

$\Theta_{1_{new}} = \Theta_i$ − Const * +Ve $= \Theta_j$ → $\Theta_{1_{new}}$ will decrease

→ we will reach to Local minimum.

② Slope = − Ve Value.



* $\Theta_{1_{old}} = \Theta_i$

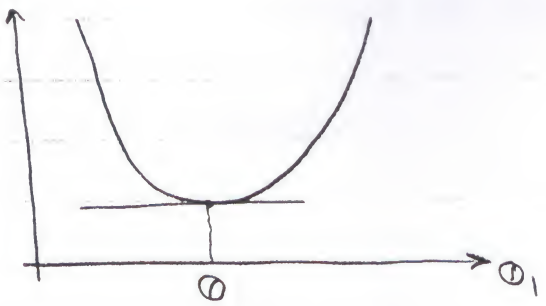
$\Theta_{1_{new}} = \Theta_i - (\text{const} * -\text{Ve}) = \Theta_i + (\text{const} * +\text{Ve}) = \Theta_j$

→ $\Theta_{1_{new}} = \Theta_j$ will increase

→ we will reach to Local minimum.

Note : • the Value of Slope doesn't effect reaching to Local min it only effects the value of Θ_1 by increase or decrease.

→ at Local minimum.



$$\Theta_{1\,new} = \Theta = \Theta_{1\,old} - \alpha \frac{d J(\Theta_1)}{d \Theta_1}$$

at this point $\frac{d J(\Theta_1)}{d \Theta_1} = \text{Zero}$

∴ $\Theta_{new} = \Theta_{old}$ ∴ Θ_1 doesn't change

→ We reach to the Final Point when $J(\Theta_1)$, which depends on the value change of Θ_1, doesn't change
"Here we reach to Local minimum"

Note: the Slope Change changes the theta Θ_1 value (increase, decrease) but doesn't change our way to reach Local minimum.

Gradient Descent For Linear Regression With One Variable.

→ Here we will study how to use gradient descent algorithm to get the Best Linear regression model $h_\theta(x) = \theta_0 + \theta_1 x$ by minimizing the objective Function $J(\theta_0, \theta_1)$

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$$

→ Gradient descent algorithm.

repeat until Convergence {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

(For $j=1$ & $j=0$)

}

to use gradient descent algorithm to get best Linear regression we need to get $\frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$ to be able to use this algorithm.

→ Gradient Descent Algorithm For Linear Regression :-

repeat until Convergence {

$$\theta_0 := \theta_0 - \alpha \underbrace{\frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)}$$

$\frac{\partial J(\theta_0, \theta_1)}{\partial \theta_0}$ حيث θ_1 ثابتة

$$\theta_1 := \theta_1 - \alpha \underbrace{\frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right) x^{(i)}}$$

$\frac{\partial J(\theta_0, \theta_1)}{\partial \theta_1}$ حيث θ_0 ثابتة

}

→ Update θ_0 and θ_1 Simultaneously

θ_0 ، θ_1 تتغيران في تزامن كما سبق الشرح قيمة

→ Using MatLab to drow GDA for linear regression we get the drow in slide 17

We Note that it has one Local minimum point (global minimum)

SO, This mean that IF we start from any value OF θ_0, θ_1 we reach to the same point

by another meaning :-

IF we start from known values of θ_0, θ_1 for the first time and start from different values of θ_0, θ_1 from th first time for the second time and repeate starting from different values for every time we reach to the same Local minimum point

.Here, we have one Local minimum point we can called it global minimum.

Local minimum $\equiv$ Global minimum.

"Batch" Gradient Descent

Batch : Each step of gradient descent uses all the training examples.